ch:4

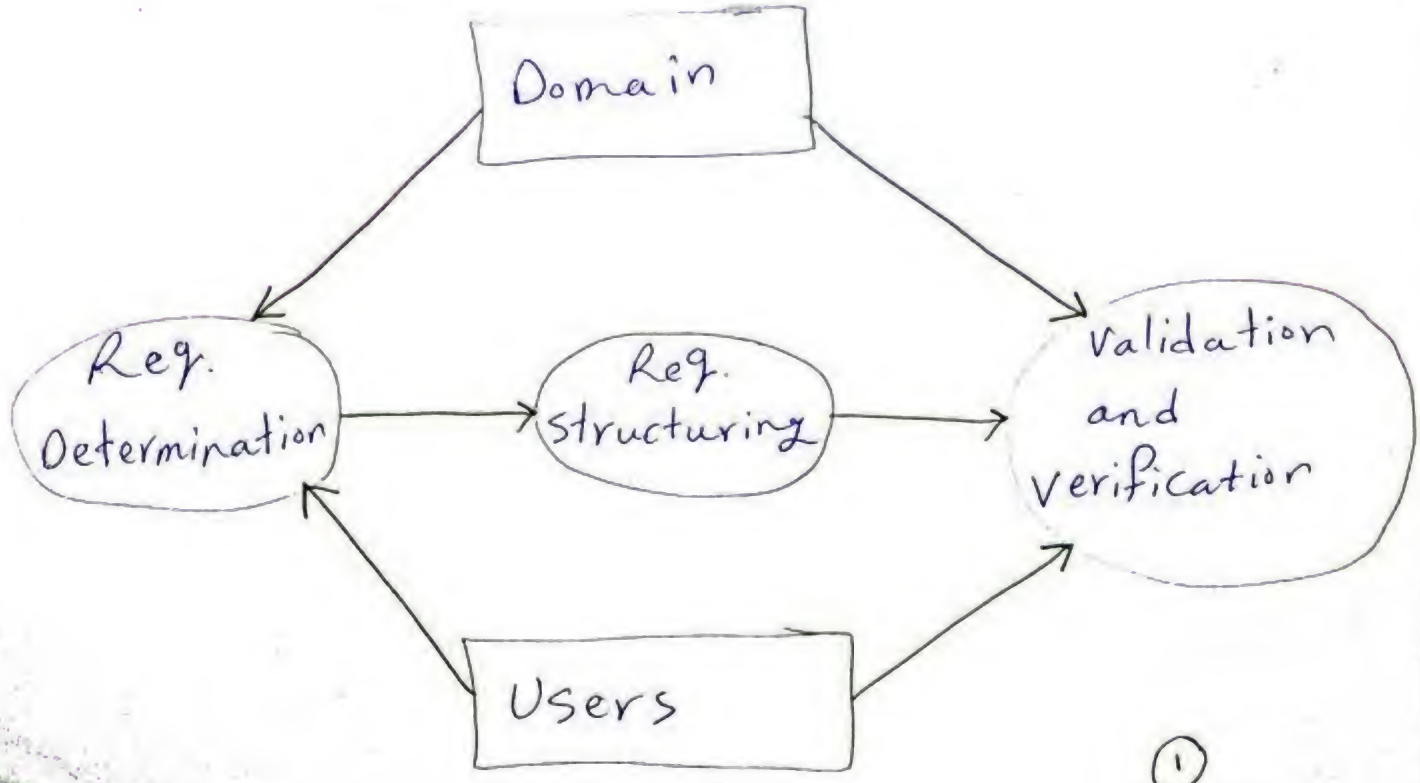
Determining system requirements

* Requirement engineering Process

→ Process of determinging the services that the customer requires from a system and the constraints under which it operates and is developed.

Requirements

→ description of the services and constraints that are generated during requirement engineering process



[1] Performing Requirements Determination (Elicitation)

* Gather information on what system should does from many soukos.

→ users : interviews, questionnaire, forms ....
→ Brain storming sessions.
→ Documents (Domain) : Reports, Procedures, domain

⇒ Note make SW for specific users, elicitation is different from making Generic SW ?!

* characteristics for Gathering requirements.

1)
→ question everything (EX: are all transactions Proceeded in the same way)

2) impartiality
↳ find best organizational solution. Consider all ideas and try to find the best.

3) Relaxation of Constraints: Assume everything is possible and eliminate the infeasible

4) Attention of details.

5) Reframing view organization in new ways according to users Requirements.

Traditional Methods for determining Requirements

1) Interviewing and listening

↝ Gather facts and opininons.
↝ observe body language and emotions.

Guidelines

a) make a plan.
b) Appointment (Convenient for the interviewee)
↳ check list of the interview.
↳ Give him an idea of the subject of interview (he may need to prepare things before)
c) Be natural.
d) listen Carefully.
e) Type up notes within 48 hours.
f) do not set expectations about new system.

Interview Questions

*oper-Ended no Pre-Specified answers (give interviewee freedom to answer)

*Close-Ended: Interviewee is asked to choose from set of specified responses (T/F or MCQ or Rank)

* Do not Phrase questions in ways that imply a wrong or right answer.

[2] Questionnaires

→ More cost-effective than interviews.
→ Be careful when choosing respondents.
  ↳ should be representative of all users.

→ Design
  ↳ mostly closed-ended questions.
  ↳ Can be administered over the phone or in a person.

[3] Interviewing Groups

→ make an interview of a group instead of individually.

Advantage
a) more effective use of time.
b) Enables people to hear opinions of others and to agree or disagree.

Disadvantage
↳ Difficulty in scheduling a specific time.

[4] Directly observing users

→ observe users while they are working to understand how do they perform the work.
→ serves as a good method to supplement interviews.

Disadvantage

→ difficulty to obtain unbiased data as People often work differently when being observed (better or worse) → may be egyptians

---

Analyzing Procedures and other Documents

→ Four types of useful documents

a) written work Procedures

→ Describes how a Job is Performed.
→ Includes data and information used and created in Process of Performing the Job or task

b) Business form

↳ explicitly indicate data flow in or out of system..

c) Report

↳ Enables the analyst to work backdraws. from the report to data that generated it.

d) Description of current information system.

Types of info. to be discovered:-

a) Problems with existing system.
b) opportunity to meet new need.
c) organizational direction d) values of organization
e) Names of Key individuals. f) rules of processing data.
g) Reasons for current system design.

Modern Methods for Determining Requirements

1) Joint application Design (JAD)

→ Brings together Key users, managers and system analysts.

→ Purpose: Collect system requirements together from Key people.

~~READ~~ L→ End Result of JAD

→ Documentation detailing existing system.

→ Features of proposed system.

2) Prototyping

a) Repetitive Process. b) Replaces or arguments SDLC.

c) Rudimentary version of system is built.

Goal to develop Concrete specifications for ultimate system.

*Most useful when

a) user requests or not clear.

b) Few users are involved in the system.

c) Tools are readily available to build prototype.

d) Designs are complex and require Concrete form.

*Drawbacks

a) Difficult to adapt to more general user audience.

b) Tendency to avoid formal documentation.

c) sharing data with other systems is often not considered.

(V & V) Verification and Validation

1) Verification

a) are we building the things right ? (way of building)

b) we wrote every details, no errors in spelling, or something missing.

2) Validation

a) are we building the things right? (the thing itself)

b) is each requirement written as it should be programmed or is it clear (not understandable)

Requirements (Functional & non-functional)

1) Functional requirements

→ describes functions requried, how system should react to particular ilps and how system should behave.

2) Non-Functional

→ Define system properties and constraints.

→ Specify particular programming language or development method.

→ Non-functional requirements may be more critical than functional requirements → if these are not met, system is useless.

→ There is a classification for non-functional requirements at slide4.pdf.

Examples of non-functional requirements

Property	Measure
Speed	→ Processed transaction per second. → user/Event response time. → screen refresh time.
size	→ K-bytes. → no. of RAM chips
Ease of use	→ Training time. → Number of help frames.
Reliability	→ Mean time of failure. → Probability of unavailability. → Availability.
Robustness	→ time to restart after failure. → Probability of data corruption on failure.
Probability.	→ Percentage of target dependent statements. → no. of target systems

Non-functional requirements

1) Product requirements

↳ specify that delivered product must behave in a particular way.

ex: execution speed, portability, reliability ..etc.

2) Organizational requirements

→ driven from organizational policies and procedures.

ex: process standards used, implementation requirements as colours, fonts used, no. of users delivery date ..etc.

3) External requirements

→ It is arised from factors which are external to the system.

ex: interoperability, requirements with other programs --etc.

Requirement document

chls of Good System analyst

(1) impertinence

→ you should ask many questions to collect informations ~~about~~ v. ask about any thing not clear.

(2) impartiality

→ you must find best organizational solution to a business problem. don't impose yourself, examine all ideas and take a decision.

(3) Relaxation of Constraints

→ assume that every thing is possible, eliminate the infeasible (ex. not applicable)

(4) Attention of details

→ take notes of every detail said on interview or written in a form.

(5) Reframing :-

→ you mus not Jump to Conclusion about built system. (b) view system as users requirements tell not as you have seen it before in prev. system.

Requirement specification

→make the requirement that will be used in design and as a guide to show to the users.

Types of requirement document

1) user requirements "written for customer"
→statements written in natural language and diagrams of services the system provides.

2) system requirements
→written as a contract between client & contractor.
→A structured document setting out detailed descriptions of system services.

3) Software specification "written for developers"
→ A detailed software description which is used for design or implementation.

How to write requirement document

Part 1 : Intro
→It describe the SW to be built in brief giving it all details of SW such as (functions of sw, goal of building system....)

Part 2 Hardware

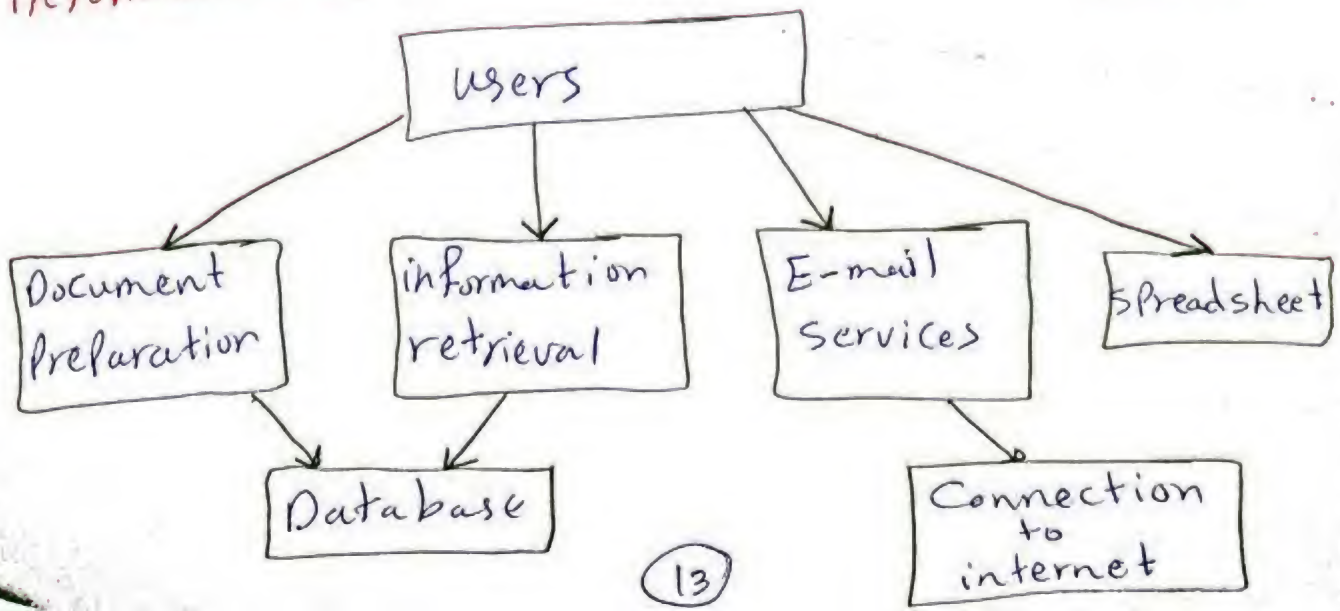
→ Describe the HW which SW will run on.

→ if using on-self HW, it describes the min. configuration that SW will work under (RAM size, Processor speed ----)

part 3 : Conceptual model

→ graphical representation of the SW services required, their relationship, any external HW or SW related to these services.

Ex: if Company need SW that perform the following functions, Document Preparation (word processor), information retrieval (Database), Emailing, spreadsheet (Excel)



→ in Prev. Page : each of these services Can be described in more details in separated figures.

→ The no. of figures of conceptual model of sw implies the size of sw and also its complexity.

Part 4: Functional requirements

→ it fully describes the functions (services) required in sw and drawn in conceptual model.

Functional user Requirement: may be high-level statements of what system should do.

→ but it should describe system in detail

⇒ How to write functional requirement part

[1] using Natural language (NL)

It's Problems

a) Presence of excess noise

→ Presence of parts of text that could be viewed as noise as it doesn't Contain information relative to function.

2) Precision is difficult

↳ without making the document difficult to read.

3) Requirements confusion.

4) Functional & non-functional requirements tend to be mixed up.

5) several different requirements may be expressed together.

6) It can neglect important details that is necessary to understand the function.

[2] using Formal or semi formal language.

*Formal as in Programming language. Not available till now.

*Semi-formal impose some structure on the way we should write the functional req. like templates or forms should be filled.

→ Thus, it limits the NL parts written.

⇒ some of these languages use graphs to express functions and their interrelationship.

→ using semi-formal

*Examples of templates

1) Name of function: ------
2) task to be made: ------
3) Produced from previous function: ------
4) Inputs to function: ---
5) outputs from function: -----

⇒ Examples of semi-formal language

1) PSL/PSA : Programmable structured language/ Programmable structured analysis.

2) RSL : Requirement specification language
→ It has a compiler that accept structured forms and generates NL version to show to users.

3) SADT structured analysis developing tool
↳ uses structured forms and graphical symbols.

Part 5 : Non-functional requirements:

→ Define system properties & constraints. such as reliability, response time..... etc.

→ Process req. may also specify particular Programming language or developing method

→ Non-functional may be more critical than functional . to make sys. useful.

Part 6 : Database requirements:

→ This part will contain databased requirements if system uses a database.

→ we should include entities required to be stored and relations presented by E-R model.